IN-CONTEXT LEARNING OF REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work demonstrates that structured patterns in pretraining data influence how representations of different concepts are organized in a large language model's (LLM) internals, with such representations then driving downstream abilities. Given the open-ended nature of LLMs, e.g., their ability to in-context learn novel tasks, we ask whether models can flexibly alter their semantically grounded organization of concepts. Specifically, if we provide in-context exemplars wherein a concept plays a different role than what the pretraining data suggests, can models infer these novel semantics and reorganize representations in accordance with them? To answer this question, we define a toy "graph tracing" task wherein the nodes of the graph are referenced via concepts seen during training (e.g., `apple`, `bird`, etc.), and the connectivity of the graph is defined via some predefined structure (e.g., a square grid). Given exemplars that indicate traces of random walks on the graph, we analyze intermediate representations of the model and find that *as the amount of context is scaled, there is a sudden re-organization of representations according to the graph's structure*. Further, we find that when reference concepts have correlations in their semantics (e.g., `Monday`, `Tuesday`, etc.), the context-specified graph structure is still present in the representations, but is unable to dominate the pretrained structure. To explain these results, we analogize our task to energy minimization for a predefined graph topology, which shows getting non-trivial performance on the task requires for the model to infer a connected component. Overall, our findings indicate context-size may be an underappreciated scaling axis that can flexibly re-organize model representations, unlocking novel capabilities.

1 INTRODUCTION

A growing line of work demonstrates that large language models (LLMs) organize representations of concepts in a manner that reflects their structure in pretraining data (Engels et al., 2024; Park et al., 2024a;b; Anthropic AI, 2023; 2024); e.g., Engels et al. (2024) show that concepts such as days of the week and months of the year form a cyclical organization in the latent space. More targeted experiments in synthetic domains have corroborated such findings as well; e.g., Li et al. (2022) use toy board games and show that LLMs can form world representations that mirror the underlying board state. These organized representations have been argued to underlie and influence a model's capabilities as well (Anthropic AI, 2023; 2024; Rimsky et al., 2024). However, as a model interacts with the world, we expect it to exhibit the ability to learn about novel concepts on-the-fly. Currently, users address this challenge by exploiting the open-ended nature of LLMs and directly specifying *in-context* the novel definition of a concept (Qin et al., 2023; Bubeck et al., 2023; Brown et al., 2020b; Agarwal et al., 2024; Anil et al., 2024). However, one can easily expect such novel concepts to not align with the structures internalized by the model during pretraining. For example, assume we describe in-context to an LLM that a new corporate enterprise called *Strawberry* has been announced—does the model sufficiently "understand" that we are referring to a corporate entity and not the fruit "strawberry"?

Motivated by the above, we design a toy task that helps evaluate whether when provided an in-context specification of a concept, an LLM alters its representations to reflect the specified task-relevant semantics, overriding the one internalized during pretraining. In particular, our proposed task involves a simple "graph tracing" problem, wherein the model is shown edges corresponding to a random traversal of a graph. The nodes of this graph are intentionally referenced via concepts the

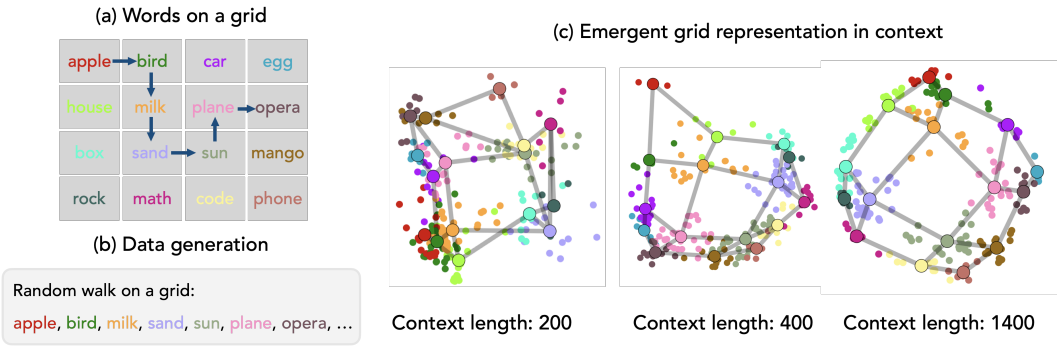

Figure 1: **Formation of an in-context task representation mirroring a grid structure.** (a) We randomly arrange a set of tokens on a grid structure that do not reflect their semantics. (b) We then generate sequences of tokens following a random walk on the grid which is used as the input context to the model. (c) The model's mean token representations projected onto the top two principal components. As the number of in-context exemplars increases, there is a formation of representations mirroring the grid structure underlying the data-generating process. The representations are from the residual stream activation after layer 26.

model is extremely likely to have seen during training (e.g., `apple`, `bird`, etc.), while its connectivity structure is defined using a predefined geometry (e.g., a square grid). Based on the provided context, the model is expected to output a valid next node prediction, i.e., a node connected to the last presented one. *As we show, increasing the amount of context leads to a sudden re-organization of representations in accordance with the graph's connectivity.* These results suggest LLMs can manipulate their representations in order to reflect information specified *entirely in-context*, enabling flexibility in their downstream use. To explain these results, we draw a connection to the problem of energy minimization on a graph, finding that achieving non-trivial accuracy on our task requires for the model to identify connected components of the graph (and hence its structure). This result also yields a connection to prior works studying the problem of bond percolation on a graph, motivating us to find the effect of graph size scaling on our results. Interestingly, we find the critical amount of context needed to solve our task scales as a power law whose exponents are well-aligned with the ones predicted in percolation theory. We thus hypothesize the model implicitly performs a percolation-like process to yield the results observed in our experiments. Overall, our contributions can be summarized as follows.

- **Graph Navigation as a Simplistic Model of Novel Semantics.** We introduce a toy graph navigation task that requires a model to interpret semantically meaningful concepts as referents for nodes in a structurally constrained graph. Inputting traces of random walks on this graph into an LLM, we analyze whether the model alters its intermediate representations for referent concepts to predict valid next nodes as defined by the underlying graph connectivity.
- **Emergent In-Context Reorganization of Concept Representations.** Our results show that as context-size is scaled, i.e., as we add more exemplars in context, there is a sudden re-organization of concept representations that reflects the graph's connectivity structure. The context-specified graph structure emerges even when we use concepts that have correlations in their semantics (e.g., `Mon`, `Tues`, etc.), but, interestingly, is unable to dominate the pretrained structure. More broadly, we note that the sudden reorganization observed in our results is reminiscent of emergent capabilities in LLMs when other relevant axes, e.g., compute and model size, are scaled (Wei et al., 2022; Srivastava et al., 2022; Lubana et al., 2024)—our results indicate context can be deemed as yet another, and in fact a more efficient, axis for unlocking model capabilities.
- **An Energy Minimization Model of Structure Inference.** We propose an energy minimization model for our proposed task as a hypothesis for the mechanism employed by an LLM to reorganize representations according to the graph's structure. This model also draws a connection to the theory of graph percolation, based on which we analyze the implication of graph size scaling and, interestingly, find the critical context size needed for performing our task (for specific graph structures) scale in accordance with percolation theory.

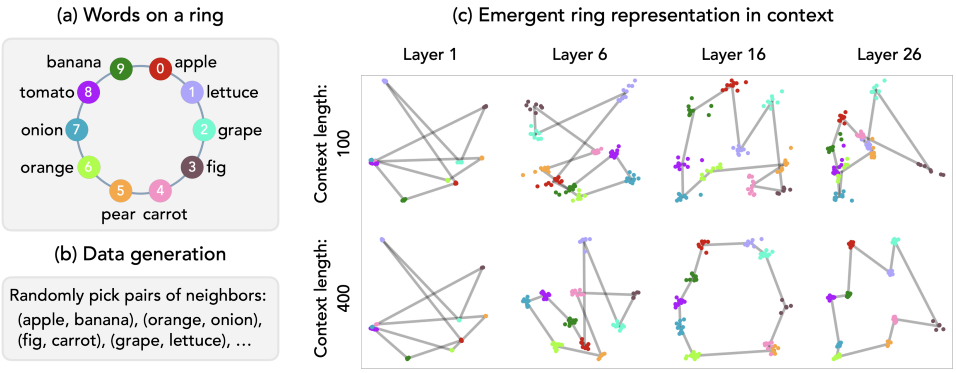

Figure 2: **Formation of an in-context task representations mirroring a ring structure.** (a) We randomly place words on a ring structure unrelated to their semantic meanings. (b) We then generate sequences of tokens by randomly sampling *neighboring pairs* from the ring which is used as the input context to the model. (c) The model's mean representation of words projected onto the top two principal components. As the number of in-context exemplars increases, there is a formation of representations mirroring the ring structure underlying the data-generating process. The representations are from the residual stream activation after layer 26.

2 Experimental Setup: In-Context Graph Tracing

We first define our setup for assessing the impact of context specification on how a model organizes its representations. In the main paper, we primarily focus on Llama3.1-8B (henceforth Llama3) (Dubey et al., 2024), accessed via NDIF/NNsight (Fiotto-Kaufman et al., 2024). We present results on other models (Llama3.2-1B, Llama3.1-8B-Instruct, Gemma-2-2B, Gemma-2-9B) in App. C.2.

Task. Our proposed task, which we call *in-context graph tracing*, involves random walks on a predefined graph $\mathcal{G}$. Specifically, inspired by prior work analyzing structured representations learned by sequence models, we experiment with three graphical structures: a square grid (Fig. 1 (a); Li et al. (2022)), a ring (Fig. 2 (a); Engels et al. (2024)), and a hexagonal grid. Results for the hexagonal grid are mostly deferred to appendix due to space constraints. To construct a square grid, we randomly arrange the set of tokens in a 4×4 grid and add edges between horizontal and vertical neighbors. We then perform a random walk on the graph, emitting the visited tokens as a sequence (Fig. 1 (b)). For a ring, we add edges between neighboring nodes and we simply sample random pairs of neighboring tokens on the graph (Fig. 2 (b)). Nodes in our graphs, denoted $\mathcal{T} = \{\tau_0, \tau_1, \ldots, \tau_n\}$, are referenced via concepts that the model is extremely likely to have seen during pretraining. While any choice of concepts is plausible, we select random tokens that, unless mentioned otherwise, have no obvious semantic correlations with one another (e.g., `apple`, `sand`, `math`, etc.). However, these concepts have precise meanings associated with them in the training data, necessitating that to the extent the model relies on the provided context, the representations are morphed according to the in-context graph. We also note that our proposed task is similar to ones studied in literature on in-context RL, wherein one provides exploration trajectories in-context to a model, expecting it to understand the environment and its dynamics (aka a world model) (Lee et al., 2024b; Laskin et al., 2022). We also highlight that a visual analog of our task, wherein one uses images instead of text tokens to represent a concept, has been used to elicit very similar results with humans as the ones we report in this paper using LLMs (Garvert et al., 2017; Whittington et al., 2020; Mark et al., 2020; 2024).

3 Results

3.1 Visualizing internal activation using principal components

Since we are interested in uncovering context-specific representations, we input sequences from our data-generating process to the model and first compute the mean activations for each unique token $\tau \in \mathcal{T}$. Namely, assume a given context $\mathcal{C} := [c_0, ..., c_{N-1}]$, where $c_i \in \mathcal{T}$, that originates from an underlying graph $\mathcal{G}$. At each timestep, we look at a window of N_w (=50) preceding tokens (or

all tokens if the context length is smaller than N_w), and collect all activations corresponding to each token $\tau \in \mathcal{T}$ at a given layer ℓ. We then compute the mean activations per token, denoted as $\boldsymbol{h}_\tau^\ell \in \mathbb{R}^d$. We further denote the stack of mean token representations as $\boldsymbol{H}^\ell(\mathcal{T}) \in \mathbb{R}^{n \times d}$. Finally, we run PCA on $\boldsymbol{H}^\ell(\mathcal{T})$, and use the first two principal components to visualize model activations (unless stated otherwise). We note that while PCA visualizations are known to suffer from pitfalls as a representation analysis method, we provide a thorough quantitative analysis in Sec. 4 to demonstrate that the model re-organizes concept representations according to the in-context graph structure, and prove in Sec. 5 that the structure of the graph is reflected in the PCA visualizations *because* of this re-organization of representations. We also provide further evidence on the faithfulness of PCA as a tool for our analysis by conducting a preliminary causal analysis of the principal components, finding that intervening on concept representations' projections along these components affects the model's ability to accurately predict valid next node generations (App. C.4).

Results. Fig. 1 (c) and Fig. 2 (c) demonstrate the resulting visualizations for square grid and ring graphs, respectively (more examples are provided in the Appendix; see Fig. 9, 10). Strikingly, with enough exemplars, we find representations are in fact organized in accordance with the graph structure underlying the context. Interestingly, results can be skewed in the earlier layers in accordance with semantic priors the model may have internalized during training; however, these priors are overridden as we go deeper in the model. For example, in the ring graph (see Fig. 2), concepts `apple` and `orange` are closer to each other in Layer 6 of the model, but become essentially antipodal around layer 26, as dictated by the graph; the antipodal nature is also more prominent as context length is increased.

We also observe that despite developing a square-grid structure when sufficient context length is given (see Fig. 1), the structure is partially irregular; e.g., it is wider in the central regions, but narrowly arranged in the periphery. We find this to be an artifact of frequency with which a concept is seen in the context. Specifically, concepts that are present in the inner 2×2 region of the grid are more frequently visited during a random walk on the graph, while the periphery of the graph has a lower visitation frequency. The representations reflect this, thus organizing in accordance with both structure and frequency of concepts in the context.

Overall, the results above indicate that *as we scale context size, models can re-organize semantically unrelated words to form in-context task-specific representations*.

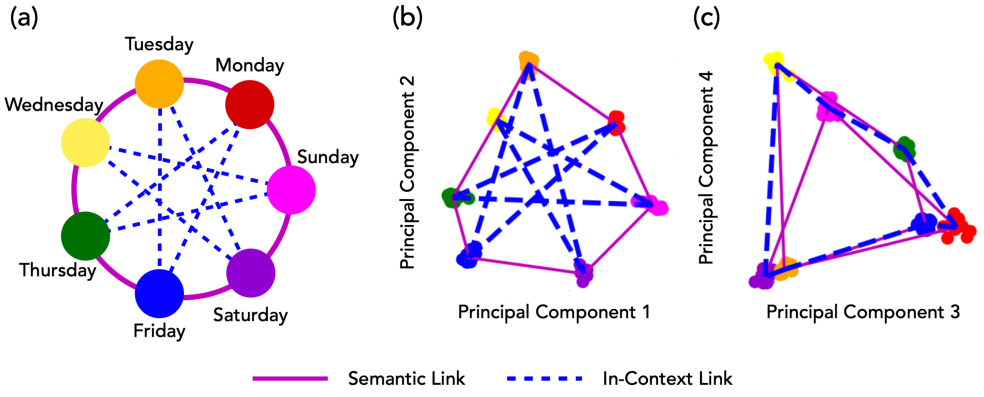

Figure 3: **In-context representations form in higher principal components in the presence of semantic priors.** (a) (Purple) Semantic links underlying days of the week. (Dashed blue) We define a non-semantic graph structure by linking non-neighboring days and generate tokens from this graph. (b) (Purple) The ring geometry formed by semantic links established during pre-training remains intact in the first two principal components. (c) (Dashed blue) The non-semantic structure provided in-context can be seen in the third and fourth principal components. Note that the star structure in the first two components (b), which match the ground truth graphical structure of our data generating process (a), becomes a ring in the next two principal components (c). The representations are from the residual stream activation after layer 21.

3.2 Semantic Prior vs. In-Context Task Representations

Building on results from the previous section, we now investigate the impact of using semantically correlated tokens. Specifically, we build on the results from Engels et al. (2024), who show that representations for days of the week, i.e., tokens {`Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday`}, organize in a circular geometry. We randomly permute the ordering of these tokens, arrange them on a 7-node ring graph similar to the previous section, and evaluate whether we can override the strong pretraining prior internalized by the model.

Results. Fig. 3 (b, c) demonstrate the resulting visualizations. We find that when there is a conflict between the semantic prior and in-context task, we observe the *original semantic ring* in the first two principal components. However, the components right after in fact encode the context-specific structure: visualizing the third and fourth principal components shows the newly defined ring structure. This indicates that the context-specified structure is present in the representations, but not dominating them. In App. C.6 Fig. 14, we report the model's accuracy on the in-context task finding that the model overrides the semantic prior to perform well on the task when enough context is given.

4 Effects of Context Scaling: Emergent Re-Organization of Representations

Our results in the previous section demonstrate models can re-organize concept representations in accordance context-specified structures. We next aim to study how this behavior arises as context is scaled—is there a continuous, monotonic improvement towards the context-specified structure as context is added? If so, is there a trivial solution, e.g., regurgitation based on context that helps explain these results? To analyze these questions, we must first define a metric that helps us gauge how aligned the representations are with the structure of the graph that underlies the context.

Dirichlet Energy. We measure the *Dirichlet energy* of our graph $\mathcal{G}$'s structure by defining an energy function over the model representations. Specifically, for an undirected graph $\mathcal{G}$ with n nodes, let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ be its adjacency matrix, and $\boldsymbol{x} \in \mathbb{R}^n$ be a signal vector that assigns a value x_i to each node i. Then the Dirichlet energy of the graph with respect to $\boldsymbol{x}$ is defined as

$$E_{\mathcal{G}}(\boldsymbol{x}) = \sum_{i,j} \boldsymbol{A}_{i,j}(x_i - x_j)^2. \tag{1}$$

For high-dimensional signals, the Dirichlet energy is defined as the summation of the energy over each dimension. Specifically, let $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ be a matrix that assigns each node i with a d-dimensional vector $\boldsymbol{x}_i$, then the Dirichlet energy of $\boldsymbol{X}$ is defined by

$$E_{\mathcal{G}}(\boldsymbol{X}) = \sum_{k=1}^{d} \sum_{i,j} \boldsymbol{A}_{i,j}(x_{i,k} - x_{j,k})^2 = \sum_{i,j} \boldsymbol{A}_{i,j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2. \tag{2}$$

Overall, to empirically quantify the formation of geometric representations, we can measure the Dirichlet energy with respect to the graphs underlying our data generating processes (DGPs) and our mean token activations $\boldsymbol{h}_\tau^\ell$:

$$E_{\mathcal{G}}(\boldsymbol{H}^\ell(\mathcal{T})) = \sum_{i,j} \boldsymbol{A}_{i,j} \|\boldsymbol{h}_i^\ell - \boldsymbol{h}_j^\ell\|^2, \tag{3}$$

where $\boldsymbol{H}^\ell(\mathcal{T}) \in \mathbb{R}^{n \times d}$ is the stack of our mean token representations $\boldsymbol{h}^\ell$ at layer ℓ and $i, j \in \mathcal{T}$ are tokens from our DGP. Intuitively, the measure above indicates whether neighboring tokens (nodes) in the ground truth graph have a small distance between their representations. *Thus, as the model correctly infers the correct underlying structure, we expect to see a decrease in Dirichlet energy.* We do note that, in practice, Dirichlet energy minimization has a trivial solution where all nodes are assigned the same representation. While we can be confident this trivial solution does not exist in our results, for else we would not see distinct node representations in PCA visualizations nor high accuracy for solving our tasks, we still provide an alternative analysis in App. C.3 where the representations are standardized to render this trivial solution infeasible. We find qualitatively similar results with such standardized representations, but more noisy since standardization can induce sensitivity to noise.

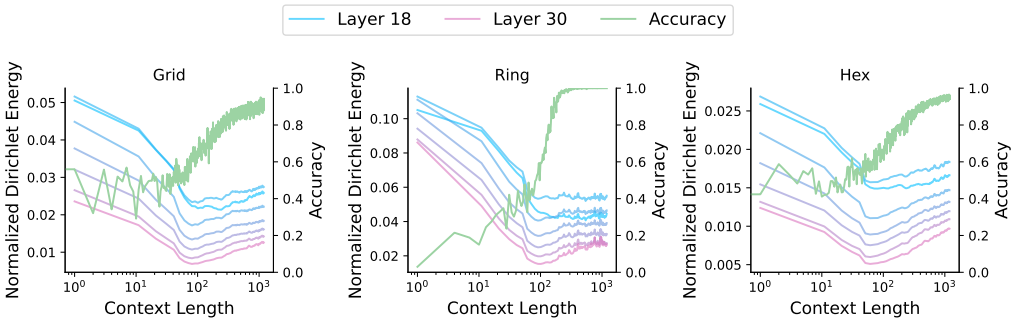

Figure 4: **A model continuously develops task representation as it learns to traverse novel graphs in-context.** We plot the accuracy of graph traversal and the Dirichlet energy of the graph, computed from the model's internal representations, as functions of context length. We note that the Dirichlet energy never reaches a perfect zero—ruling out that the representations are learning a degenerate structure, as was also seen in the PCA visualizations in Sec. 3. (a) A 4x4 grid graph with 16 nodes. (b) A circular ring with 10 nodes. (c) A "honey-comb" hexagonal lattice, with 30 nodes.

4.1 Results: Emergent Organization and Task Accuracy Improvements

We plot Llama3's accuracy at the in-context graph tracing task alongside the Dirichlet energy measure (for different layers) as a function of context. Specifically, we compute the "rule following accuracy", where we add up the model's output probability over all graph nodes which are valid neighbors. For instance if the graph structure is `apple-car-bird-water` and the current state is `car`, we add up the predicted probabilities for `apple` and `bird`. This metric simply measures how well the model abides by the graph structure, irrelevant of its accuracy.

Results are reported in Fig. 4. We see once a critical amount of context is seen by the model, accuracy starts to rapidly improve. We find this point in fact closely matches when Dirichlet Energy reaches its minimum value: energy is minimized shortly before the rapid increase in in-context task accuracy, suggesting that the structure of the data is correctly learned before the model can make valid predictions. This leads us to the claim that *as the amount of context is scaled, there is an emergent re-organization of representations that allows the model to perform well on our in-context graph tracing task*. We note these results also provide a more quantitative counterpart of our PCA visualization results before.

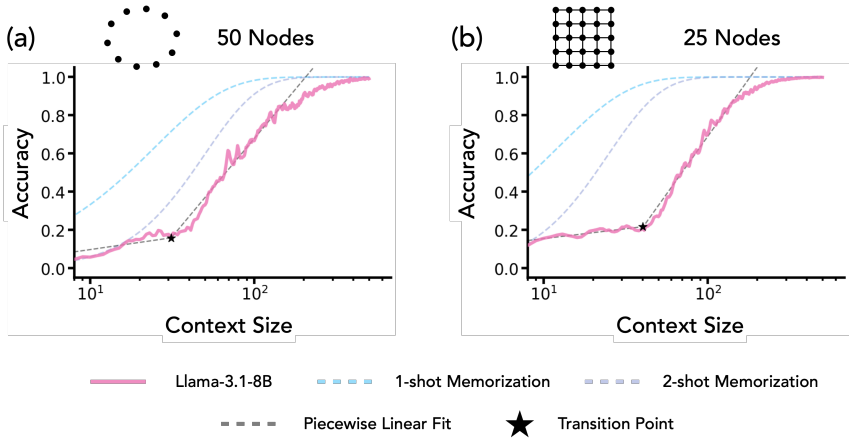

Figure 5: **A memorization solution cannot explain Llama's ICL graph tracing performance.** We plot the rule following accuracy from Llama-3.1-8B outputs and accuracies from a simple 1-shot and 2-shot memorization hypothesis. (a) A ring graph with 50 nodes. (b) A square grid graph with 25 nodes. In both cases, we find that the memorization solution cannot explain the accuracy ascent curve. Instead, we find a slow phase and a fast phase, which we fit with a piecewise linear fit.

Is there a Trivial Solution at play? An simple solution which can drive the accuracy to increase is when the model is merely regurgitating a node's neighbors by copying them from its context. We call this the *memorization solution*. Since our accuracy metric measures rule following, this memorization solution will achieve value 1 if the node has been observed in the context and 0 otherwise. Here, we investigate whether this solution is plausible by sampling data using the previously described random sampling strategy for both the grid and the ring. Since this sampling procedure simply chooses an initial node at random with replacement, we can express the probability of a node existing in a context of length l as:

$$p_{\text{seen1}}(\mathbf{x}) = 1 - \left(\frac{n-1}{n}\right)^{l}, \tag{4}$$

where $\mathbf{x}$ is the context and n is the number of nodes available. Note that the current node itself does not matter as the sampling probability is uniform with replacement. Since one might expect that a language model needs to encounter the same token twice to recognize it as an in-context exemplar, we also define the probability that a node appeared twice as:

$$p_{\text{seen2}}(\mathbf{x}) = p_{\text{seen1}}(\mathbf{x}) - \binom{l}{1}\left(\frac{1}{n}\right)^{1}\left(\frac{n-1}{n}\right)^{(l-1)}. \tag{5}$$

Where $\binom{l}{1} = l$ is the binomial coefficient. To evaluate whether this hypothesis explains our results, we plot these two memorization solutions with the observed performance of Llama-3. Fig. 5 shows the result (a) on a ring graph with 50 nodes and (b) on a grid graph with 25 nodes. We find, in both cases, that neither the 1-shot memorization curve nor the 2-shot memorization curve can explain the behavior of Llama. Instead, we observe that the accuracy has two phases, a first phase where the accuracy improves very slowly and a second phase where the log-linear slope suddenly changes to a steeper ascent. We find that a piecewise linear fit can extract this transition point robustly, which will be of interest in the next section.

5 Explaining emergent re-organization of representations: The energy minimization hypothesis

In this section, we put forward a hypothesis for why we are able to identify such structured representations from model internals: the model internally runs an *energy minimization process* in search of the correct structural representation of the data. More formally, we claim the following hypothesis.

Hypothesis 5.1. *Let n be the number of tokens, d be the dimensionality of the representations, and $\boldsymbol{H}(\mathcal{T}) \in \mathbb{R}^{n\times d}$ be the stack of representations for each token learned by the model, then $\mathbb{E}_{\mathcal{G}}(\boldsymbol{H}((\mathcal{T}))$ decays with context length.*

Minimizers of Dirichlet Energy and Spectral Embeddings. We call the k-th energy minimizer of $E_{\mathcal{G}}$ the optimal solution that minimizes $E_{\mathcal{G}}$ and is orthogonal to the first $k-1$ energy minimizers. Formally, the energy minimizers $\left\{\boldsymbol{z}^{(k)}\right\}_{k=1}^{n}$ are defined as the solution to the following problem:

$$\boldsymbol{z}^{(k)} = \arg\min_{\boldsymbol{z}\in\mathbb{S}^{n-1}} E_{\mathcal{G}}(\boldsymbol{z}) \tag{6}$$

$$\text{s.t.} \quad \boldsymbol{z} \perp \boldsymbol{z}^{(j)}, \forall j \leq k-1, \tag{7}$$

where $\mathbb{S}^{n-1}$ is the unit sphere in n dimensional Euclidean space.

The energy minimizers are known to have the following properties (Spielman, 2019):

1. $\boldsymbol{z}^{(1)} = c\mathbf{1}$ for some constant $c \neq 0$, which is a degenerated solution that assigns the same value to every node; and
2. If we use $\left(z_i^{(2)}, z_i^{(3)}\right)$ as the coordinate of node i, it will be a good planar embedding. We call them (2-dimensional) **spectral embeddings**.

Spectral embeddings are often used to a draw graph on a plane and in many cases can preserve the structure of the graph (Tutte, 1963). In Figures 6 and 7, we show the spectral embedding results for

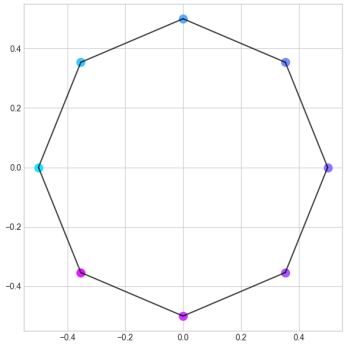

Figure 6: Spectral embedding of a ring graph.

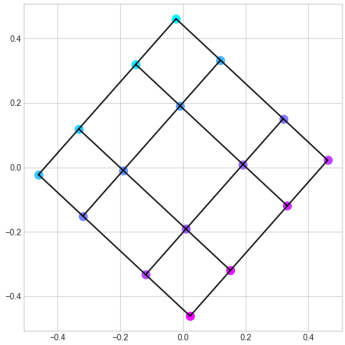

Figure 7: Spectral embedding of a grid graph.

a ring graph and a grid graph respectively. Notice how such spectral embeddings are similar to the representations from our models in Fig. 1 and 2.

Most importantly, we prove in Theorem B.1 that, if the representations $\boldsymbol{H}$ from the model are minimizing the Dirichlet energy and is non-degenerated, then the first two principal components of PCA will exactly produce the spectral embeddings $\boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}$. Here we present an informal version of the theorem, and defer the full version and proof to the appendix.

Theorem 5.1 (Informal Version of Theorem B.1)**.** *Let $\mathcal{G}$ be a graph and $\boldsymbol{H} \in \mathbb{R}^{n \times d}$ (where $n \geq d \geq 3$) be a matrix that minimizes Dirichlet energy on $\mathcal{G}$ with non-degenerated singular values, then the first two principal components of $\boldsymbol{H}$ will be $\boldsymbol{z}^{(2)}$ and $\boldsymbol{z}^{(3)}$.*

See App. B for the formal version and proof of Theorem 5.1, and Tab. 2 for an empirical validation.

Connectivity and Energy Minimization Given the relationship between spectral embeddings (i.e., energy minimizers) and the principal components that we observe (Figures 1, 2), we hypothesize that the model's inference of the underlying structure is akin to energy minimization. If this hypothesis is true, then we can further make connections to studies of graph percolation to predict when we can expect emergent behavior from scaling context, a process that can be analogized to filling in edges in a graph (i.e., the bond percolation sub-problem in percolation theory) (Newman, 2003; Hooyberghs et al., 2010). Specifically, we expect to see emergent behavior for our in-context tasks once the model correctly infers a *large connected component* of the underlying graph, after having observed a sufficient number of exemplars in the context. For the scenario of lattice structures with square and hexagonal grids, prior work has shown the percolation transition point scales as a power law with exponents of 0.5 and 0.65 respectively (Wikipedia., 2024). Rather fascinatingly, we find these exponents match the scaling exponents retrieved from experiments with Llama3 models! We defer details of these experiments to Sec. 5.1, and discuss the significance of a connected component to finish this section.

Namely, we demonstrate that the moment at which we can visualize a graph using PCA implies the moment at which the model has found a large connected component. Consider an unconnected graph $\hat{\mathcal{G}}$, i.e. $\hat{\mathcal{G}}$ has multiple connected components. Then there will be multiple degenerate solutions to the energy minimization, which will be found by PCA. Specifically, suppose $\hat{\mathcal{G}}$ has q connected components, with the node set of the i-th component being U_i. Then we can construct the first q energy minimizers in the following way: for any $i \in [q]$, let the j-th value of $\boldsymbol{z}^{(i)}$ be

$$z_j^{(i)} = \begin{cases} -\alpha_i & j \in \bigcup_{k=1}^{i-1} U_k \\ 1 & \text{otherwise,} \end{cases} \tag{8}$$

where $\alpha_1 = 1$ and $\alpha_i = \frac{\sum_{k'=i}^{q} \sum_{j \in U_{k'}} z_{j'}^{(i-1)}}{\sum_{k=1}^{i-1} \sum_{j \in U_k} z_j^{(i-1)}}$ for $i \in [q] \setminus \{1\}$.

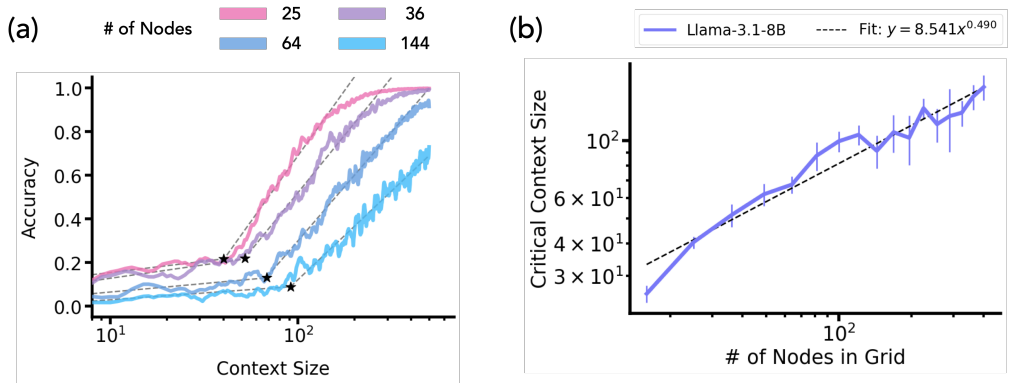

Figure 8: **A Percolation transition could explain in-context emergence.** We analyze the in-context accuracy curves based on percolation theory. The graph used in this experiment is a $m \times m$ grid where we vary m. (a) The rule following accuracy of a graph tracing task. The accuracy show a two phase ascent. We fit a piecewise linear function to the observed ascent to extract the transition point. (b) We plot the context size at the transition point as a function of n, the number of nodes in the graph. We find 0.490, similar to the 0.5 exponent expected from percolation theory.

This way, it is easy to check that each $\boldsymbol{z}^{(i)}$ constructed above for $i \in [q]$ has 0 energy, thus is a global minimizer of $E_{\hat{\mathcal{G}}}$. Moreover, all $\boldsymbol{z}^{(i)}$'s are orthogonal to each other. Therefore, they satisfy our definition of the first q energy minimizers.

It is important to notice that $\boldsymbol{z}^{(i)}$'s above for $i \in [q]$ contain no information about the structure of the graph other than identifying each connected component. Theorem B.1 tells us that the principal components of a non-degenerated (rank s where $s > 1$) solution $\boldsymbol{H}$ that minimizes the energy will be $\boldsymbol{z}^{(2)} \cdots \boldsymbol{z}^{(s+1)}$. Thus, if the graph is unconnected, then the energy-minimizing representations will be dominated by information-less principal components, in which we should not expect any meaningful visualization.

The acute reader may remember that the first minimizer $\boldsymbol{z}^{(1)}$ is a trivial solution of the energy minimization that assigns the same value to every node. Conveniently, this is not a concern, as PCA will rule out this degenerate solution as demonstrated in Theorem B.1.

5.1 A Percolation Transition under Graph-Size Scaling?

Building on the relation between largest connected component and the bond percolation phase transition suggested above, we now evaluate whether empirical results on whether the critical amount of context-size needed for achieving non-trivial accuracy matches the predicted power-law scaling from percolation theory.

Results. We are interested in observing how critical transition points (notated T_c) scale with respect to graph size. To this point, we repeat our experiments from Section 2, but with varying numbers of nodes in our underlying graph $\mathcal{G}$. This results in a set of accuracy curves, each of which demonstrate a similar trajectory as that of Fig. 4. Given the consistent discontinuity in all of the resulting accuracy curves, we then derive the critical transition points for each run by fitting bilinear piecewise splines with a single knot that maximally explain each accuracy curve. The knots thus indicate the critical transition points for each run. Thus we are able to derive critical transition points for varying degrees of graph size (i.e., number of nodes per graph). The results for our square grid task are provided in Fig. 8, with more plots available in Appendix. The exponents identified in prior work on bond percolation in a square and hexagonal grid graph argue that we should see a power scaling with an exponent of 0.5 and 0.65, respectively (Wikipedia., 2024). The empirical results align well with these expected exponents, as shown in Figs. 8, 19.

6 Related Work

Model Representations. Researchers have recently discovered numerous representations in neural networks. Mikolov et al. (2013) suggests that concepts are *linearly* represented in activations,

and Park et al. (2024b) more recently suggests this may be the case for contemporary language models. Numerous researchers have found concrete examples of linear representations for human-level concepts, including "truthfulness" (Marks & Tegmark, 2024; Burns et al., 2022; Li et al., 2023b), "refusal" (Arditi et al., 2024), toxicity (Lee et al., 2024a), sycophancy (Rimsky et al., 2024), or even "world models" (Li et al., 2022; Nanda et al., 2023). Park et al. (2024a) finds that hierarchical concepts are represented with a tree-like structure consisting of orthogonal vectors. A relevant line of work includes that of Todd et al. (2023) and Hendel et al. (2023). Both papers find that one can compute a vector from in-context exemplars that encode the task, such that adding such a vector during test time for a new input can correctly solve the task. Language models do not always form linear representations, however. Engels et al. (2024) find circular feature representations for periodic concepts, such as days of the week or months of the year, using a combination of sparse autoencoders and PCA. Csordás et al. (2024) finds that recurrent neural networks trained on token repetition can either learn an "onion"-like representation or a linear representation, depending on the model's width. *Unlike such prior work, we find that task-specific representations with a desired structural pattern can be induced in-context.* To our knowledge, our work offers the first such investigation of in-context representation learning.

Scaling In-Context Learning Numerous works have demonstrated that in-context accuracy improves with more exemplars (Brown et al., 2020a; Lu et al., 2022). With longer context lengths becoming available, researchers have begun to study the effect of *many-shot* prompting (as opposed to few-shot) (Agarwal et al., 2024; Anil et al., 2024; Li et al., 2023c). For instance, Agarwal et al. (2024) reports improved performance on ICL using hundreds to thousands of exemplars on a wide range of tasks. Similarly, Anil et al. (2024) demonstrate the ability to jail-break LLMs by scaling the number of exemplars. Unlike such work that evaluates model behavior, *we study the effect of scaling context on the underlying representations*, and provide a framework for predicting when discontinuous changes in behavior can be expected via mere context-scaling.

7 Discussion

In this work, we show that LLMs can flexibly manipulate their representations from structures internalized based on pretraining data to structures defined entirely in-context. To arrive at these results, we propose a simple but rich task of graph tracing, wherein traces of random walks on a graph are shown to the model in-context. The graphs are instantiated using predefined structures (e.g., lattices) and concepts that are semantically interesting (e.g., to define nodes), but meaningless in the overall context of the problem. Interestingly, we find the ability to flexibly manipulate representations is in fact emergent with respect to context size—we propose a model based on energy minimization and graph percolation to hypothesize a mechanism for the underlying dynamics of this behavior. These results suggest context-scaling can unlock new capabilities, and, more broadly, this axis may have as of yet been underappreciated for improving model abilities. In fact, we note that, to our knowledge, our work is to first to investigate the formation of representations entirely in-context. Our study also naturally motivates future work towards formation of world representations Li et al. (2023a) and world models (Ha & Schmidhuber, 2018) in-context, which can have significant implications toward building general and open-ended systems as well as forecasting its safety concerns. We also highlight the relation of our experimental setup to similar tasks studied in neuroscience literature Garvert et al. (2017); Mark et al. (2020; 2024), wherein humans are shown random walks of a graph of visual concepts; fMRI images of these subjects demonstrate the formation of a structured representation of the graph in the hippocampal–entorhinal cortex, similar to our results with LLMs.

Limitations. We do emphasize that our work has a few limitations. Namely, PCA, or more broadly, low dimensional visualizations of high dimensional data can be difficult to interpret or sometimes even misleading. Despite such difficulties, we provide theoretical connections between energy minimization and principal components to provide a compelling explanation for why structures elicited via PCA faithfully represent the in-context graph structure. Second, we find a strong, but nevertheless incomplete, causal relationship between the representations found by PCA and the model's predictions. We view the exact understanding of how these representations form, and the exact relationship between the representations and model predictions as an interesting future direction, especially given that such underlying mechanism seems to depend on the scale of the context.

REFERENCES

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. Many-shot in-context learning, 2024. URL https://arxiv.org/abs/2404.11018.

Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, et al. Many-shot jailbreaking. *Anthropic, April*, 2024.

Anthropic AI. *Towards Monosemanticity: Decomposing Language Models With Dictionary Learning*, 2023. https://transformer-circuits.pub/2023/monosemantic-features.

Anthropic AI. *Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet*, 2024. https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020b.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Róbert Csordás, Christopher Potts, Christopher D Manning, and Atticus Geiger. Recurrent neural networks learn to store and generate sequences using non-linear representations. *arXiv preprint arXiv:2408.10920*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Joshua Engels, Isaac Liao, Eric J. Michaud, Wes Gurnee, and Max Tegmark. Not all language model features are linear, 2024. URL https://arxiv.org/abs/2405.14860.

Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

Jaden Fiotto-Kaufman, Alexander R Loftus, Eric Todd, Jannik Brinkmann, Caden Juang, Koyena Pal, Can Rager, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Michael Ripa, Adam Belfki, Nikhil Prakash, Sumeet Multani, Carla Brodley, Arjun Guha, Jonathan Bell, Byron Wallace, and David Bau. Nnsight and ndif: Democratizing access to foundation model internals, 2024. URL https://arxiv.org/abs/2407.14561.

Mona M Garvert, Raymond J Dolan, and Timothy EJ Behrens. A map of abstract relational knowledge in the human hippocampal–entorhinal cortex. *elife*, 6:e17086, 2017.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624.

H. Hooyberghs, B. Van Schaeybroeck, and J. O. Indekeu. Percolation on bipartite scale-free networks. *Physica A: Statistical Mechanics and its Applications*, 389(15):2920–2929, August 2010. ISSN 0378-4371.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*, 2022.

Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K Kummerfeld, and Rada Mihalcea. A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity. *arXiv preprint arXiv:2401.01967*, 2024a.

Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024b.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2022.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=DeG07_TcZvT.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=aLLuYpn83y.

Mukai Li, Shansan Gong, Jiangtao Feng, Yiheng Xu, Jun Zhang, Zhiyong Wu, and Lingpeng Kong. In-context learning with many demonstration examples. *arXiv preprint arXiv:2302.04931*, 2023c.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556.

Ekdeep Singh Lubana, Kyogo Kawaguchi, Robert P Dick, and Hidenori Tanaka. A percolation model of emergence: Analyzing transformers trained on a formal language. *arXiv preprint arXiv:2408.12578*, 2024.

Shirley Mark, Rani Moran, Thomas Parr, Steve W Kennerley, and Timothy EJ Behrens. Transferring structural knowledge across cognitive maps in humans and models. *Nature communications*, 11 (1):4783, 2020.

Shirley Mark, Phillipp Schwartenbeck, Avital Hahamy, Veronika Samborska, Alon B Baram, and Timothy E Behrens. Flexible neural representations of abstract structural knowledge in the human entorhinal cortex. *Elife*, 13, 2024.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets, 2024. URL https://arxiv.org/abs/2310.06824.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/abs/1301.3781.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*, 2023.

M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, January 2003. ISSN 0036-1445, 1095-7200.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024a.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models, 2024b. URL https://arxiv.org/abs/2311.03658.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828.

Daniel Spielman. Spectral and algebraic graph theory. *Yale lecture notes, draft of December*, 4:47, 2019.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

William Thomas Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 3 (1):743–767, 1963.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

James CR Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263, 2020.

Wikipedia. *Percolation Threshold*, 2024. https://en.wikipedia.org/wiki/Percolation_threshold.

A Additional Experimental Details

Here we provide some additional details regarding our experimental setups.

Context Windows. Our analyses require computing mean token representations $\boldsymbol{h}_i$ for every token $i \in \mathcal{T}$ in our graphs. To do so, we grab the activations per each token in the most recent context window of N_w tokens. Because we further require that each token is observed at least once in our window, we use a batch of prompts, where the batch size is equal to the number of nodes in our graph. For each prompt in the batch, we start our random traversal (or random pairwise sampling) with a different node, ensuring that each node shows up at least once in the context. In the case when our context length (N_c) is longer than the window, we simply use every token ($N_w = N_c$).

Computational Resources. We run our experiments on either A100 nodes, or by using the APIs provided by NDIF (Fiotto-Kaufman et al., 2024).

Code Release. We will release the code for all of our experiments after the peer review process.

B The Connection Between Energy Minimization and PCA Stucture

In this section, for a matrix $\boldsymbol{M} \in \mathbb{R}^{n\times d}$, we use lower case bold letters with subscript to represent the columns for $\boldsymbol{M}$, e.g. $\boldsymbol{m}_k$ represents the k-th column of $\boldsymbol{M}$. Moreover, we use $\sigma_k(\boldsymbol{M})$ to represent the k-th largest singular value of $\boldsymbol{M}$ and when $\boldsymbol{M}$ is PSD we use $\lambda_k(\boldsymbol{M})$ to represent the k-th largest eigenvalue of $\boldsymbol{M}$. Moreover, we use $\boldsymbol{e}_k$ to represent a vector with all-zero entries except a 1 at entry k, whose dimension is inferred from context, and $\mathbf{1}$ to represent a vector with all entries being 1. For a natural number n, we use $[n]$ to represent $\{1, 2, \cdots, n\}$.

In this section we use $\left\{\boldsymbol{z}^{(k)}\right\}_{k=1}^{n}$ to represent the k-the energy minimizers of the Dirichlet energy, defined in Section 4. Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be the adjacency matrix of the graph, $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{A}\mathbf{1})$ be the degree matrix, and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ be the Laplacian matrix. Through an easy calculation one can know that for any vector $\boldsymbol{x} \in \mathbb{R}^n$,

$$E_{\mathbb{G}}(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{L}\boldsymbol{x} \rangle . \tag{9}$$

Therefore, from the Spectral Theorem (e.g. Theorem 2.2.1 in Spielman (2019)), we know that $\boldsymbol{z}_k$ is the eigenvector of $\boldsymbol{L}$ corresponding to $\lambda_{n-k+1}(\boldsymbol{L}) = E_{\mathbb{G}}(\boldsymbol{z}_k)$.

We will show that, if a matrix $\boldsymbol{H} \in \mathbb{R}^{n\times d}$ minimizes the energy and is non-degenerated (have several distinct and non-zero singular values), then the PCA must exactly give the leading energy minimizers, starting from $\boldsymbol{z}_2$.

Theorem B.1. *Let $\mathbb{G}$ be a graph and $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_s > 0$ be $s \leq \min\{n, d\} - 1$ distinct positive numbers. Let matrix $\boldsymbol{H} \in \mathbb{R}^{n\times d}$ be the solution of the following optimization problem:*

$$\boldsymbol{H} = \arg \min_{\boldsymbol{X}\in\mathbb{R}^{n\times d}} E_{\mathbb{G}}(\boldsymbol{X}) \tag{10}$$

$$\text{s.t.} \quad \lambda_k(\boldsymbol{X}) \geq \epsilon_k, \ \forall k \in [r], \tag{11}$$

then the k-th principle component of $\boldsymbol{H}$ (for $k \in [r]$) will be $\boldsymbol{z}_{k+1}$.

Proof. We first prove that the leading left-singular vectors of $\boldsymbol{H}$ are exactly energy minimizers. Let $r = \min\{n, d\}$. Let the SVD of $\boldsymbol{H}$ be $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{\Sigma} = \operatorname{diag}[\sigma_1, \sigma_2, \cdots, \sigma_d]$ are the singular values of $\boldsymbol{H}$, and $\boldsymbol{U} \in \mathbb{R}^{n\times r}$, $\boldsymbol{V} \in \mathbb{R}^{r\times d}$.

Let $\boldsymbol{h}_i'$ represents the i-th row of $\boldsymbol{H}$. Notice that

$$E_{\mathbb{G}}(\boldsymbol{H}) = \sum_{i,j} \boldsymbol{A}_{i,j} \left\| \boldsymbol{h}_i' - \boldsymbol{h}_j' \right\|^2 \tag{12}$$

$$= \sum_{i,j} \boldsymbol{A}_{i,j} \left\| (\boldsymbol{e}_i - \boldsymbol{e}_j)^\top \boldsymbol{H} \right\|^2 \tag{13}$$

$$= \sum_{i,j} \boldsymbol{A}_{i,j} \left\| (\boldsymbol{e}_i - \boldsymbol{e}_j)^\top \boldsymbol{U}\boldsymbol{\Sigma} \right\|^2 \tag{14}$$

$$= \sum_{i,j} \sum_{k=1}^{r} \sigma_k^2 \left\langle \boldsymbol{e}_i - \boldsymbol{e}_j, \boldsymbol{u}_k \right\rangle^2 \tag{15}$$

$$= \sum_{k=1}^{r} \sigma_k^2 E_{\mathbb{G}}(\boldsymbol{u}_k). \tag{16}$$

Since σ_k's and $\boldsymbol{u}_k$'s are independent, no matter what are the values of $\boldsymbol{u}_k$, we know that each σ_k will take the smallest possible value, and from the given condition, it is $\sigma_k = \epsilon_k, \forall k \in [s]$, and $\sigma_k = 0, \forall k \in [r] \setminus [s]$.

Since $\boldsymbol{u}_k$'s are singular vectors, we have $\boldsymbol{u}_k$'s are orthogonal to each other. Using Theorem 1 in Fan (1949), we know that for any $s' \in [n]$, the minimizer of $\sum_{k=1}^{s'} E_{\mathbb{G}}(\boldsymbol{u}_k)$ is $\boldsymbol{u}_k = \boldsymbol{z}_k, \forall k \in [s']$. Therefore, it is evident that the minimizer of $\sum_{k=1}^{s} \sigma_k^2 E_{\mathbb{G}}(\boldsymbol{u}_k)$ must satisfies $\boldsymbol{u}_k = \boldsymbol{z}_k, \forall k \in [s]$, since from the above argument of σ_k's and the given condition condition we know that $\sigma_1 > \sigma_2 > \cdots > \sigma_s > 0$.

Now we have proved that $\boldsymbol{u}_k = \boldsymbol{z}_k, \forall k \in [s]$. Next we consider the output of PCA. Let $\boldsymbol{p}_k$ be the k-th principle component output by the PCA of $\boldsymbol{H}$. We know that $\boldsymbol{p}_k$ is the eigenvector of

$$\boldsymbol{C} = \widehat{\boldsymbol{H}}\widehat{\boldsymbol{H}}^\top \tag{17}$$

that corresponds to the k-th largest eigenvalue of $\boldsymbol{C}$, where $\widehat{\boldsymbol{H}} = \boldsymbol{H} - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\top \boldsymbol{H}$ is the centralized $\boldsymbol{H}$.

From the Spectral Theorem, we have

$$\boldsymbol{p}_k = \arg \max_{\substack{\boldsymbol{p} \in \mathbb{S}^{n-1} \\ \boldsymbol{p} \perp \boldsymbol{p}_i, \forall i \leq k-1}} \left\langle \boldsymbol{p}, \boldsymbol{C}\boldsymbol{p} \right\rangle. \tag{18}$$

Let $J = \operatorname{span}\{\boldsymbol{1}\}$ be the set of vectors whose every entry has the same value. Let $J^\perp$ be the subspace in $\mathbb{R}^n$ that is orthogonal to J. For a subspace K of $\mathbb{R}^n$, let $\Pi_K : \mathbb{R}^n \to \mathbb{R}^n$ be the projection operator onto K.

We have that

$$\boldsymbol{p}_1 = \arg \max_{\boldsymbol{p} \in \mathbb{S}^{n-1}} \left\langle \boldsymbol{p}, \boldsymbol{C}\boldsymbol{p} \right\rangle \tag{19}$$

$$= \arg \max_{\boldsymbol{p} \in \mathbb{S}^{n-1}} \left\langle \boldsymbol{p}, \left(\boldsymbol{I} - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\top \right) \boldsymbol{H}\boldsymbol{H}^\top \left(\boldsymbol{I} - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\top \right) \boldsymbol{p} \right\rangle \tag{20}$$

$$= \arg \max_{\boldsymbol{p} \in \mathbb{S}^{n-1}} \left\langle \Pi_{J^\perp}(\boldsymbol{p}), \boldsymbol{H}\boldsymbol{H}^\top \Pi_{J^\perp}(\boldsymbol{p}) \right\rangle \tag{21}$$

$$= \arg \max_{\substack{\boldsymbol{p} \in \mathbb{S}^{n-1} \\ \boldsymbol{p} \perp J}} \left\langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{p} \right\rangle, \tag{22}$$

which, again from Spectral Theorem, is the eigenvector of the second largest eigenvalue of $\boldsymbol{H}\boldsymbol{H}^\top$, which is $\boldsymbol{u}_2 = \boldsymbol{z}_2$. Using an induction and the same reasoning, it follows that for any $k \in [s]$, we have $\boldsymbol{p}_k = \boldsymbol{z}_{k+1}$. This proves the proposition. □

C Additional Results

C.1 Detailed Layer-wise Visualization of Representations

In Figure 9 and Figure 10 we provide additional visualizations per layer for each of our models and each of our data generating processes.

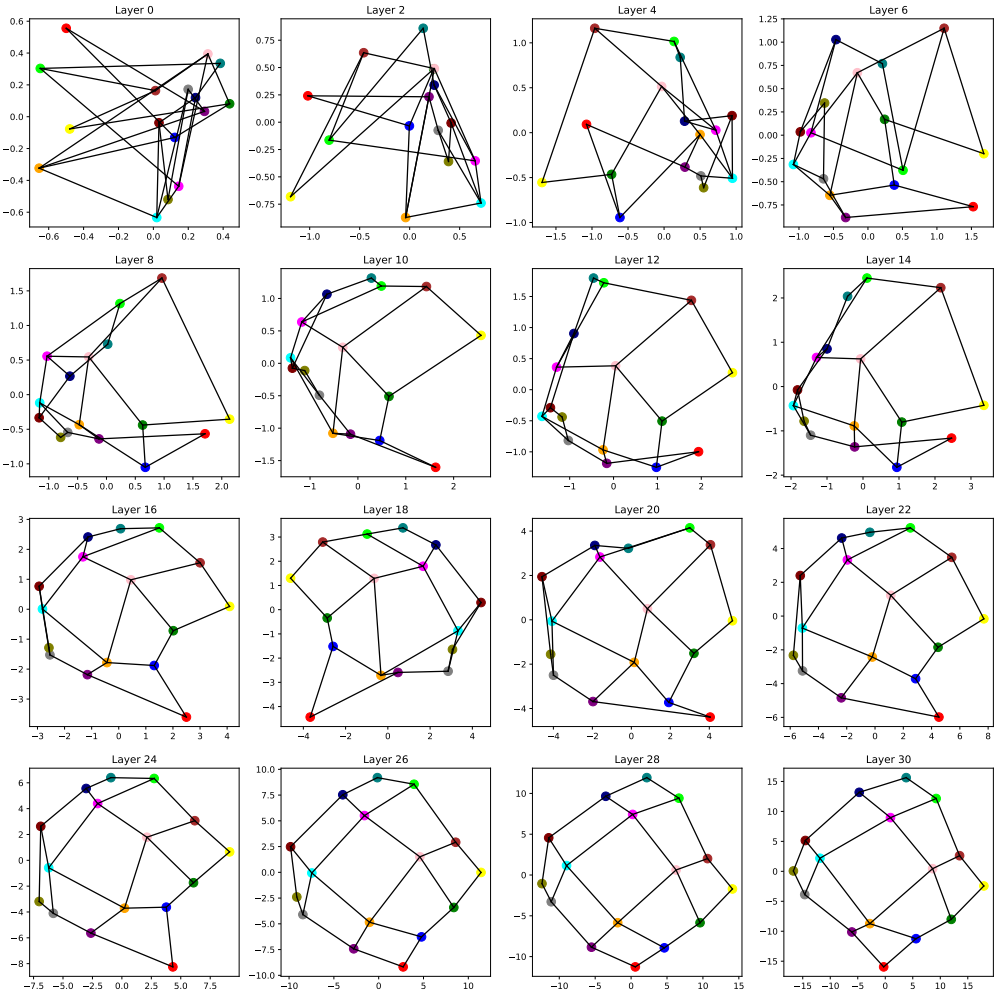

Figure 9: We plot 2d PCA projections from every other layer in Llama3.1-8B (Dubey et al., 2024) , given the board-traversal task. In deeper layers, we can see a clear visualization of the grid.

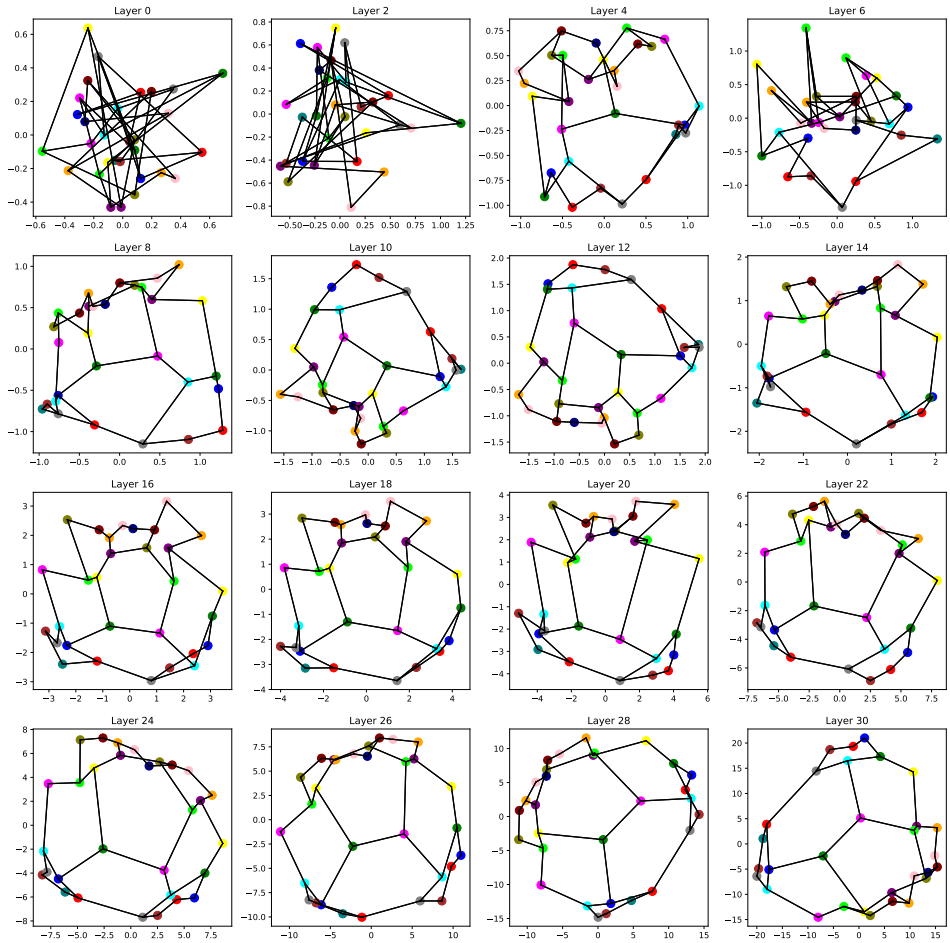

Figure 10: We plot 2D PCA projections from every other layer in Llama3.1-8B (Dubey et al., 2024) for the hexagonal grid task.

C.2 PCA, Dirichlet Energy, and Accuracy Results on Other Models

Here we provide results from other language models, i.e., Llama3-1B, Llama3-8B-Instruct, Gemma2-2B, and Gemma2-9B. In Figure 11, we plot the 2d PCA projections from the last layer of various models for various data generating processes. In Figure 12, we plot the normalized Dirichlet energy curves against accuracy for various language models on various tasks. Across all models and tasks, we see results similar to the main paper.

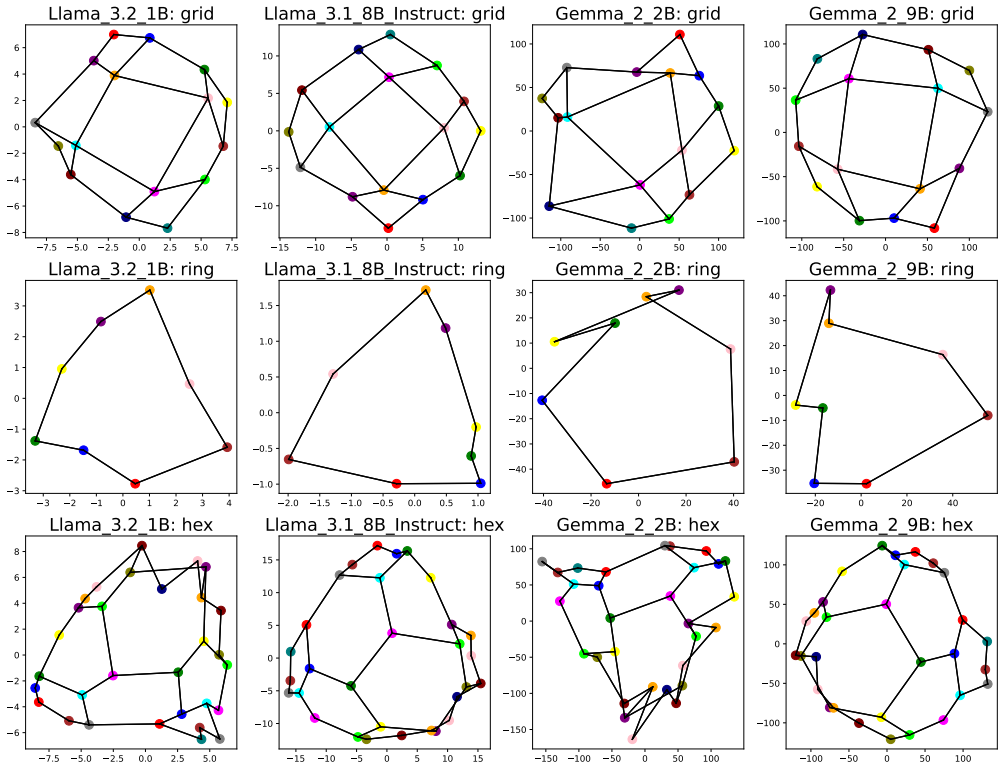

Figure 11: We plot 2d PCA projections from the last layer of various language models, given various data generating processes. For the grid and hexagonal graphs, we apply PCA on the last layers. For the rings, we visualize layers 14, 10, 16, and 20 respectively. Interestingly, for Llama3.2-1B, we find the ring representation in the 2nd and 3rd principal components.

C.3 Standardized Dirichlet Energy

In Fig. 13, we report Dirichlet energy values computed after standardization of representations. This renders the trivial solution to Dirichlet energy minimization infeasible, since assigning a constant representation to all nodes will yield infinite energy (due to zero variance). As can be seen in our results, the plots are qualitatively similar to the non-standardized energy results (Fig. 12), but more noisy, especially for the ring graphs. This is expected, since standardization can exacerbate the influence of noise, yielding fluctuations in the energy calculation.

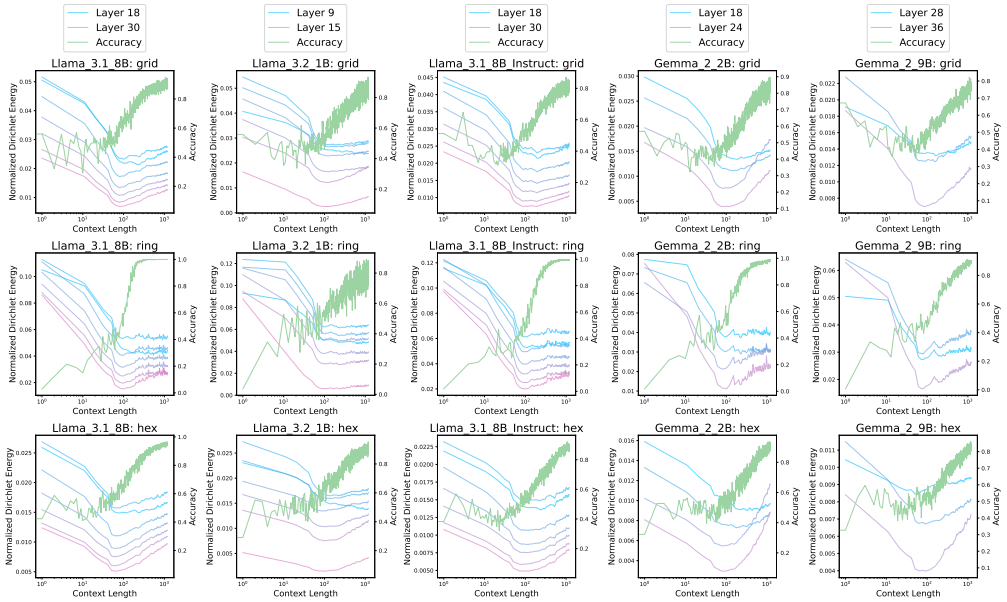

Figure 12: Accuracy versus normalized Dirichlet energy curves for various language models on various tasks. For every model and task, we see energy minimized before accuracy starting to improve.

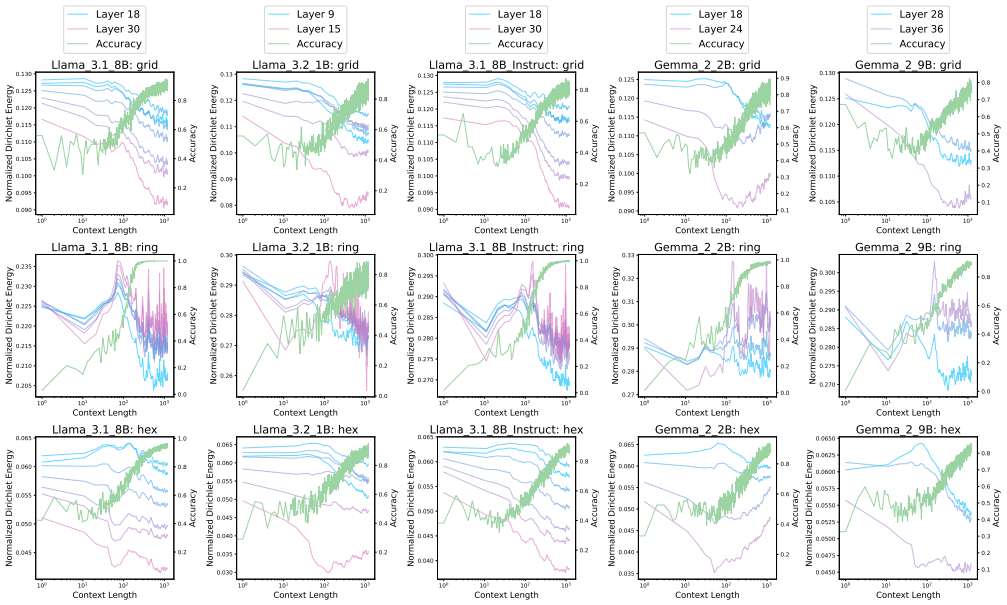

Figure 13: Accuracy versus zero mean centered normalized Dirichlet energy curves for various language models on various tasks. Zero mean centering ensures that graph representations are not using the trivial solution to energy minimization (i.e., assigning the same representation for every node).

C.4 Causal Analysis of Representations

In this section we report preliminary causal analyses of our graph representations. While fully understanding the mechanisms behind the formation of such representations, as well as the relationship between said representations and model outputs are an interesting future direction, this is not the focus of our work and thus we only ran proof-of-concept experiments.

With that said, we ask: do the principal components that encode our graph representations have any causal role in the model's predictions?

To test this, we attempt to "move" the location of the activations for one node of the graph to another by simply re-scaling its principal components. Namely, assume activation $\boldsymbol{h}_i^\ell$ corresponding to node i at layer ℓ. Say we wish to "move" the activation to a different *target node* j. We first compute the mean representation of node j using all activations corresponding to node j within the most recent N_w (= 200) timesteps, notated as $\bar{\boldsymbol{h}}_j$. Assuming the first two principal components encode the "coordinates" of the node, we simply re-scale the principal components of $\boldsymbol{h}_i$ to match that of $\bar{\boldsymbol{h}}_j$.

We view this approach as rather rudimentary. Namely, there are likely more informative vectors that encode richer information, such as information about neighboring nodes. However, we do find that the first two principal components have *some* causal role in the model's predictions.

We test our re-scaling intervention on 1,000 randomly generated contexts. For each context, assuming our underlying graph has n nodes, we test "moving" the activations of the last token i to all $n-1$ other locations in the graph. We then report the averaged metric across the resulting 1,000 $\times\ n-1$ testcases.

We report 3 metrics: accuracy (Hit@1), Hit@3, and "accumulated probability mass" on valid tokens. Hit@1 (and Hit@3) report the percentage of times at which the top 1 (top 3) predicted token is a valid neighbor of the target node j. For "accumulated probability mass", we simply sum up the probability mass allocated to all neighbors (i.e., valid predictions) of the target node j.

Table 1 reports our results for our ring and grid tasks. We include results for re-scaling with 2 or 3 principal components, as well as null interventions and interventions with a random vector. Overall, we find that the principal components have *some* causal effect on the model's output predictions, but does not provide a full explanation.

	Ring			Grid			Hex		
	Hit@1	Hit@3	Prob	Hit@1	Hit@3	Prob	Hit@1	Hit@3	Prob
Interv. (n=2)	0.61	0.91	0.6	0.57	0.95	0.55	0.30	0.32	0.69
Interv. (n=3)	0.77	0.96	0.76	0.68	0.98	0.65	0.42	0.46	0.82
Null Interv.	0.20	0.50	0.20	0.17	0.33	0.16	0.07	0.20	0.05
Random Interv.	0.17	0.47	0.19	0.16	0.37	0.17	0.06	0.18	0.05

Table 1: Intervention results for our ring and grid tasks. We demonstrate that often times, simply re-scaling the principal component for each token representation can "move" the token to a different position in the graph. However, we note that our simple re-scaling approach does not perfectly capture a causal relationship between principal components and model predictions.

C.5 Empirical Similarity of Principal Components and Spectral Embeddings

Theorem 5.1 predicts that if the model representations are minimizing the Dirichlet energy, the first two principal components will be equivalent to the spectral embeddings ($\boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}$.

Here we empirically measure whether the first two principal components are indeed equivalent to the spectral embeddings. In Table 2, we measure the cosine similarity scores between the principal components and spectral embeddings.

C.6 Accuracy of In-context tasks with a conflicting semantic prior.

What would happen when an in-context task which contradicts a semantic prior is given to a model? Namely, Engels et al. (2024) show that words like days of the week have a circular representation.

	cos_sim(PC 1, $\boldsymbol{z}^{(2)}$)	cos_sim(PC 2, $\boldsymbol{z}^{(3)}$)
Grid	0.950	0.954
Ring	0.768	0.785
Hex	0.745	0.755

Table 2: Absolute value of cosine similarity scores of principal components from model activations and spectral embeddings. We empirically observe that in practice, these coordinates end up being very similar. For the grid and hexagon, we use principal components from the last layer, while for the ring, we use an earlier layer (layer 10) in which the ring is observed.

In our experiment, we randomly shuffle tokens for days of the week (i.e., tokens {`Mon, Tue, Wed, Thu, Fri, Sat, Sun`} to define a new ring, and give random neighboring pairs from the newly defined ring as our in-context task.

Figure 14 demonstrates the accuracy when given an in-context task that is contradictory to a semantic prior. Interestingly, we first observe the model make predictions that reflects the original semantic prior (pink). This accuracy drops very quickly as the model captures that the semantic rule is not being followed. With more exemplars, we see a slow decay of the remaining semantic accuracy and a transition in the model's behavior as it begins to make predictions that reflect the newly defined ordering of our ring (blue).

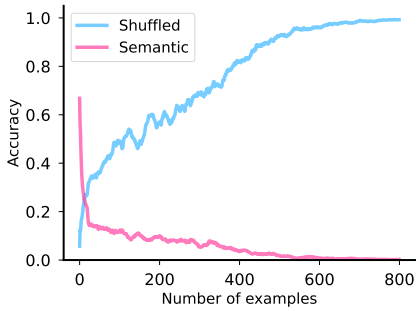

Figure 14: **In-context structure overrides semantic prior.** Given an in-context task that contradicts a model's semantic prior, we observe the model transition from making predictions that adhere to the semantic prior (pink) to predictions that reflect the newly defined in-context task.

Furthermore, in Fig. 15, we quantify the Dirichlet energy computed only from certain PC dimensions. We find that energy minimization happens in the dimensions corresponding to the in-context structure.

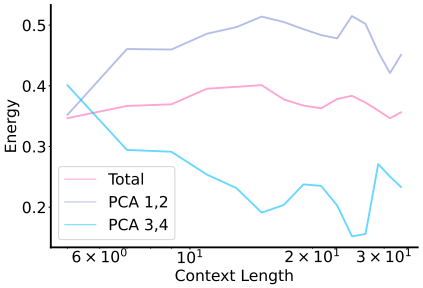

Figure 15: **Energy minimization happens in the in-context component dimensions.** We show the Dirichlet energy depending on the context given when taking 1) all 2) semantic (PCA 1,2) 3) in-context (PCA 3,4) dimensions. We show that energy minimization happens in PCA 3,4 corresponding to the in-context dimensions.

C.7 Additional Empirical Verifications of Transition Predictions

Here we provide additional details for empirically verifying our predictions for model transitions. Figures 16, 17, and 18 demonstrate detailed accuracy curves for a wide range of graph sizes.

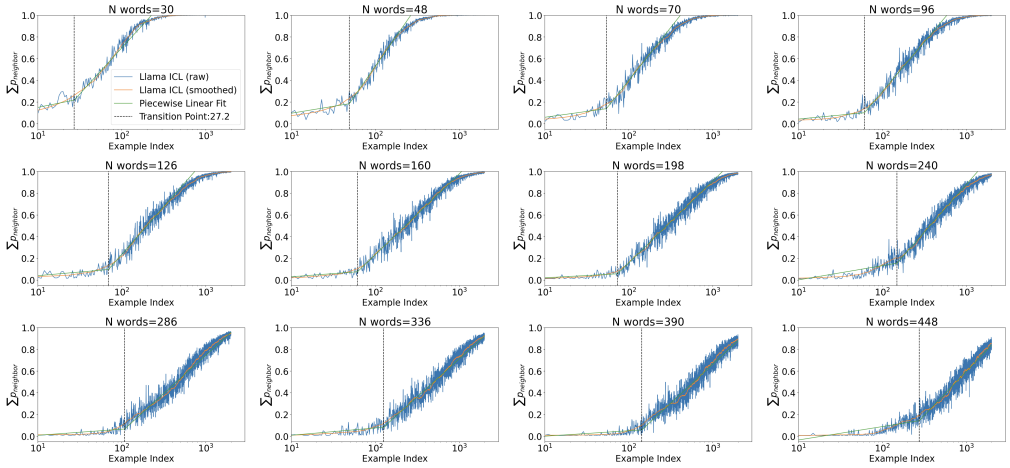

Figure 16: **Emergent behavior for varying task complexity (graph size) for the Hexagonal task.** We plot the accuracy for varying levels of complexity (graph size) for the hexagonal in-context task. Interestingly, regardless of graph size, we see an abrupt, discontinuous change in the model's performance. Figure 19 demonstrates that we can predict when such abrupt change can be expected as a function of task complexity.

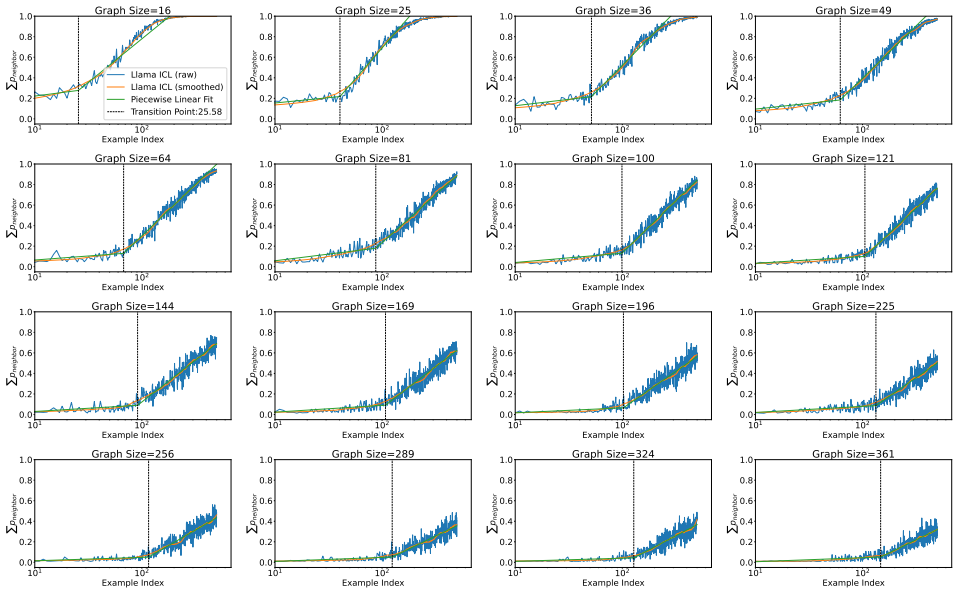

Figure 17: **Emergent behavior for varying task complexity (graph size) for the grid task.** We plot the accuracy for varying levels of complexity (graph size) for the grid in-context task. Interestingly, regardless of graph size, we see an abrupt, discontinuous change in the model's performance. Figure 8 demonstrates that we can predict when such abrupt changes can be expected as a function of task complexity.

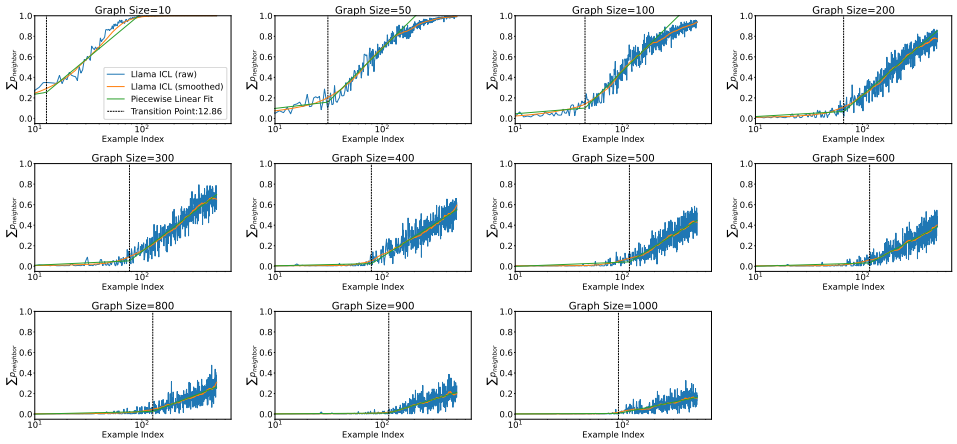

Figure 18: **Emergent behavior for varying task complexity (graph size) for the ring task.** We plot the accuracy for varying levels of complexity (graph size) for the ring in-context task. Interestingly, regardless of graph size, we again see an abrupt, discontinuous change in the model's performance.

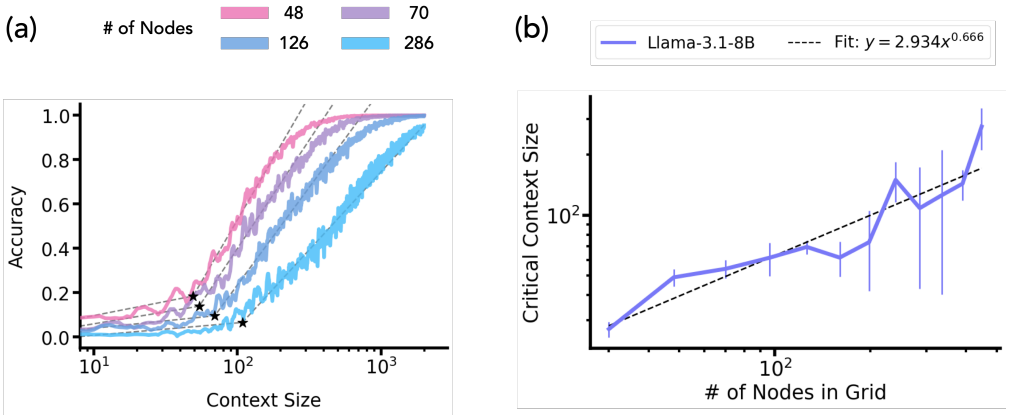

Figure 19: **Analyzing a Hexagonal graph tracing task using Percolation theory.** We analyze the in-context accuracy curves based on percolation theory. The graph used in this experiment is a $m \times m$ hexagonal grid where we vary m. (a) The rule following accuracy of the graph tracing task. The accuracy again shows a two phase ascent like Fig. 8. We fit a piecewise linear function to the observed ascent to extract the transition point. (b) We plot the context size at the transition point as a function of n, the number of nodes in the graph. We find 0.666, matching well the ~ 0.6527 exponent expected from percolation theory on a hexagonal graph.

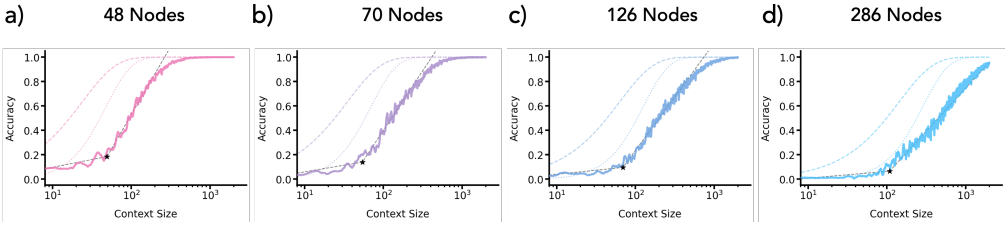

Figure 20: **Hexagonal graph tracing accuracies compared to the memorization solution** The rule following accuracies on the hexagonal graph compared to the memorization model in Sec. 4.1. Hexagonal graph with a) 48 b) 70 c) 126 d) 286 nodes. Generally we find that the hexagonal graph tracking accuracy from Llama-3.1-8B (Dubey et al., 2024) is lower than the 1,2-shot memorization model, indicating that there might be a different underlying process.